GitHub Username: raffaelcavaliere

Setlists

Description

Setlists helps you organize your music charts and makes it easy to share them with the members of your band.

Intended User

Amateur and professional musicians.

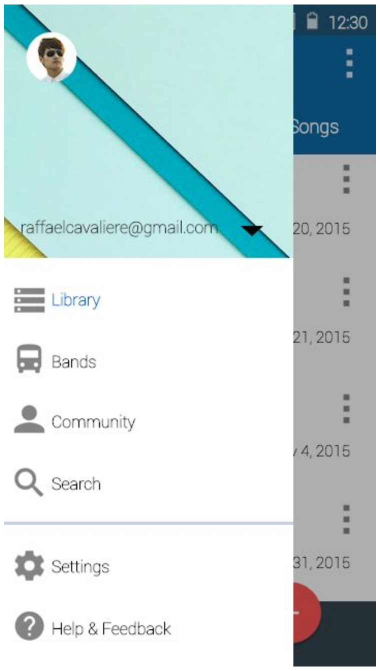
Features

- Organize songs by artist (with Spotify search)
- Add and edit songs characteristics (tempo, duration, key, etc.)
- Attach multiple documents (versions) of a song
- Supports PDF, text, image (JPEG, BMP, GIF and PNG) and chordpro formats
- Edit text and chordpro files
- Import and export documents
- Create and manage setlists (attach a visual metronome, reorder songs, etc.)
- Send MIDI messages to instruments through android midi library
- Organize bands
- Communicate with band members
- Community features (connect to other users, download their contributions)
- Authenticate through Google Sign-in and/or Firebase authentication

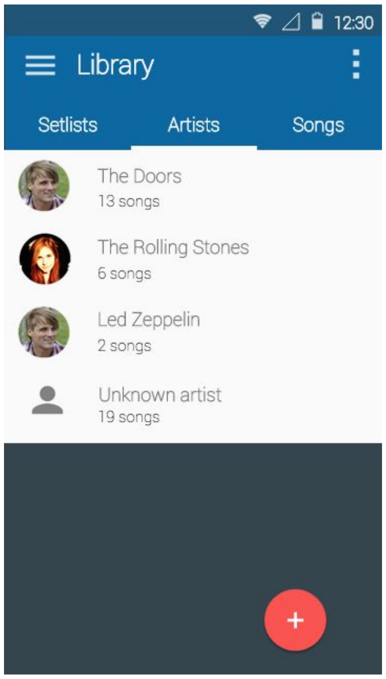
- Synchronize data between multiple devices through cloud-based repository

User Interface Mocks

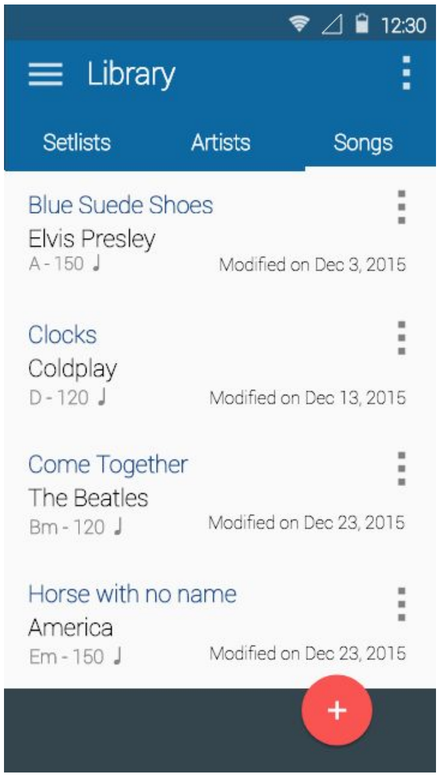
1. Drawer menu



2. Library fragment - Artists tab



3. Library fragment - Songs tab


12:30

Library

Setlists Artists Songs

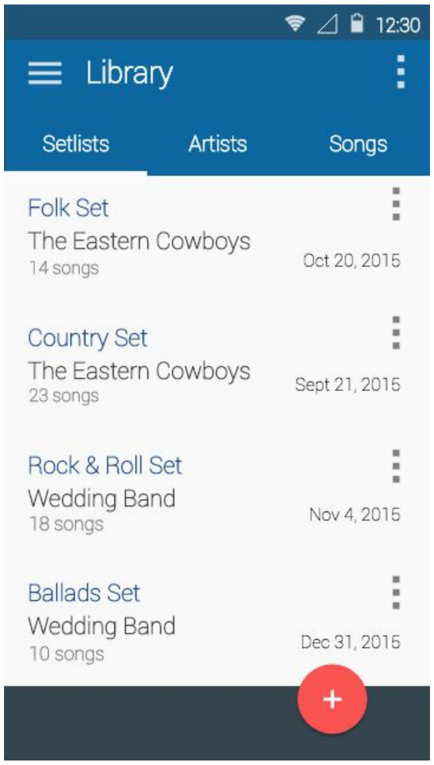
Blue Suede Shoes
Elvis Presley
A - 150 ♩ Modified on Dec 3, 2015

Clocks
Coldplay
D - 120 ♩ Modified on Dec 13, 2015

Come Together
The Beatles
Bm - 120 ♩ Modified on Dec 23, 2015

Horse with no name
America
Em - 150 ♩ Modified on Dec 23, 2015

4. Library fragment - Setlists tab


12:30

Library

Setlists Artists Songs

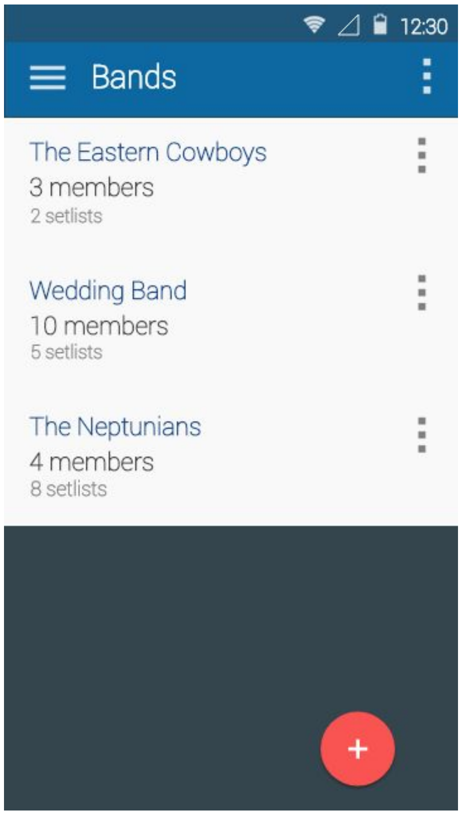
Folk Set
The Eastern Cowboys
14 songs Oct 20, 2015

Country Set
The Eastern Cowboys
23 songs Sept 21, 2015

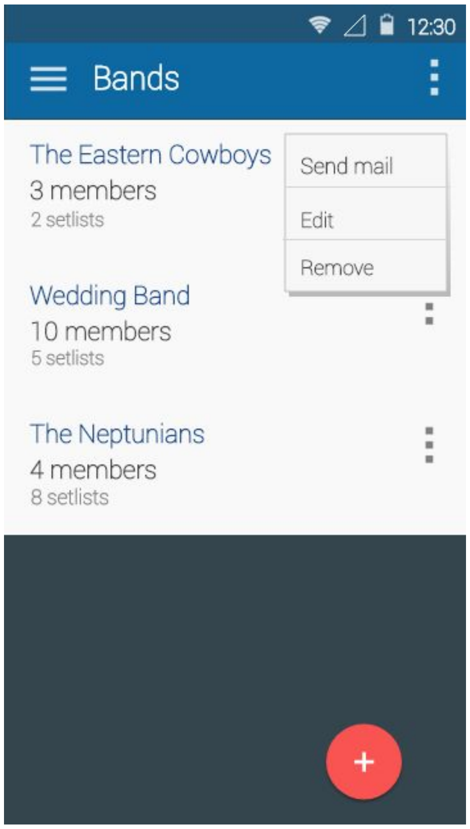
Rock & Roll Set
Wedding Band
18 songs Nov 4, 2015

Ballads Set
Wedding Band
10 songs Dec 31, 2015

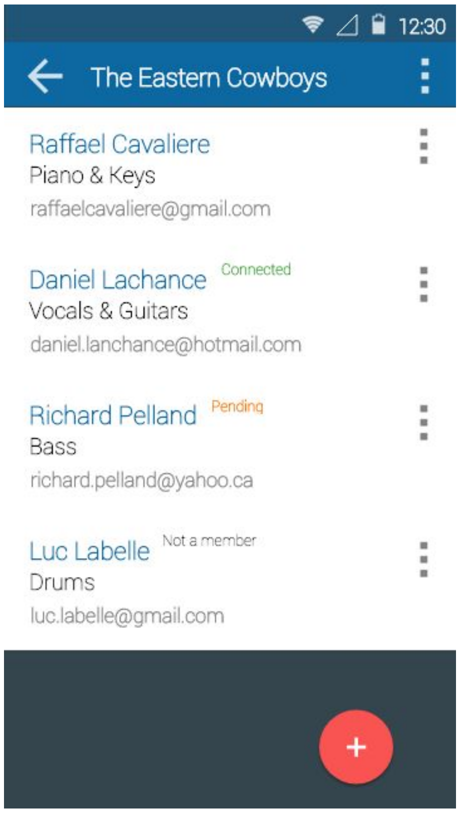
5. Bands fragment



6. Band context menu



7. Band edit



12:30

The Eastern Cowboys

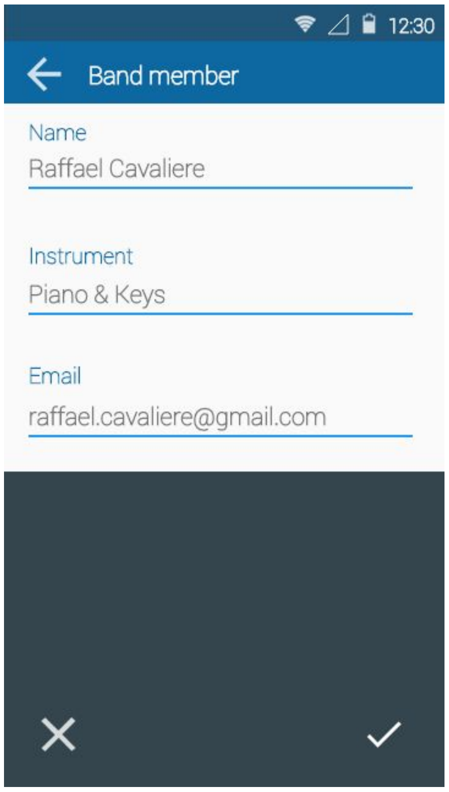
Raffael Cavaliere
Piano & Keys
raffaelcavaliere@gmail.com

Daniel Lachance Connected
Vocals & Guitars
daniel.lanchance@hotmail.com

Richard Pelland Pending
Bass
richard.pelland@yahoo.ca

Luc Labelle Not a member
Drums
luc.labelle@gmail.com

8. Band member edit



12:30

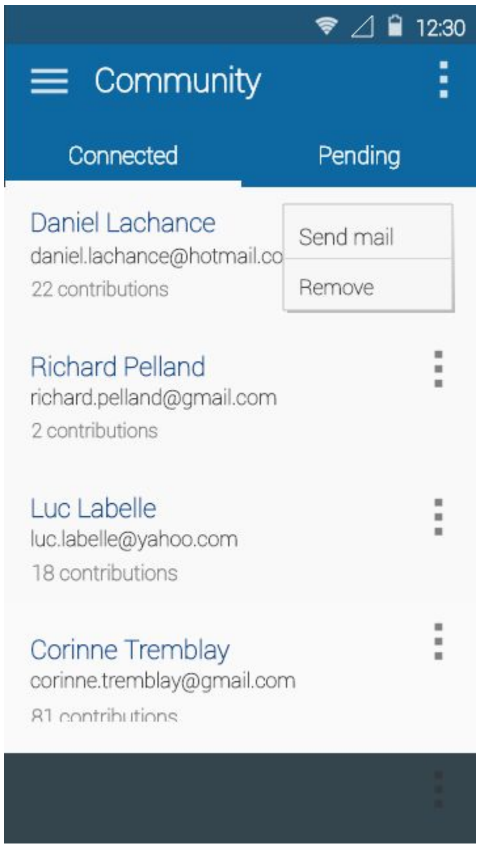
Band member

Name
Raffael Cavaliere

Instrument
Piano & Keys

Email
raffael.cavaliere@gmail.com

9. Community fragment - Connected users



12:30

Community

Connected | Pending

Daniel Lachance
daniel.lachance@hotmail.co
22 contributions

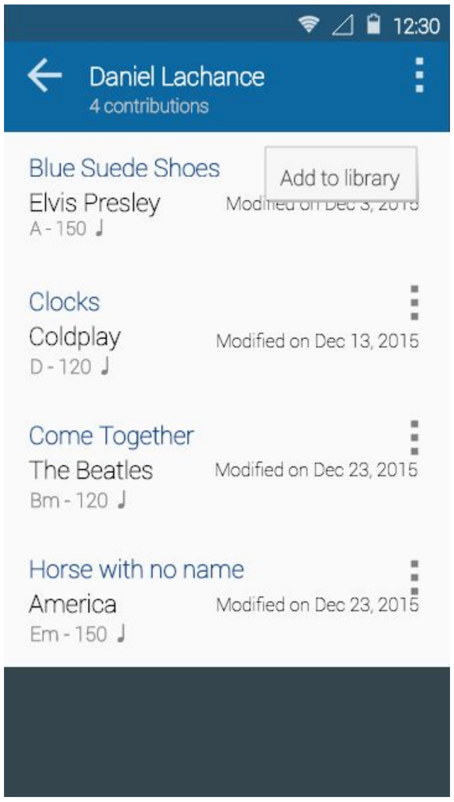
Send mail
Remove

Richard Pelland
richard.pelland@gmail.com
2 contributions

Luc Labelle
luc.labelle@yahoo.com
18 contributions

Corinne Tremblay
corinne.tremblay@gmail.com
81 contributions

10. Community fragment - View and download songs



12:30

Daniel Lachance
4 contributions

Blue Suede Shoes
Elvis Presley
A - 150 ♩
Modified on Dec 3, 2015

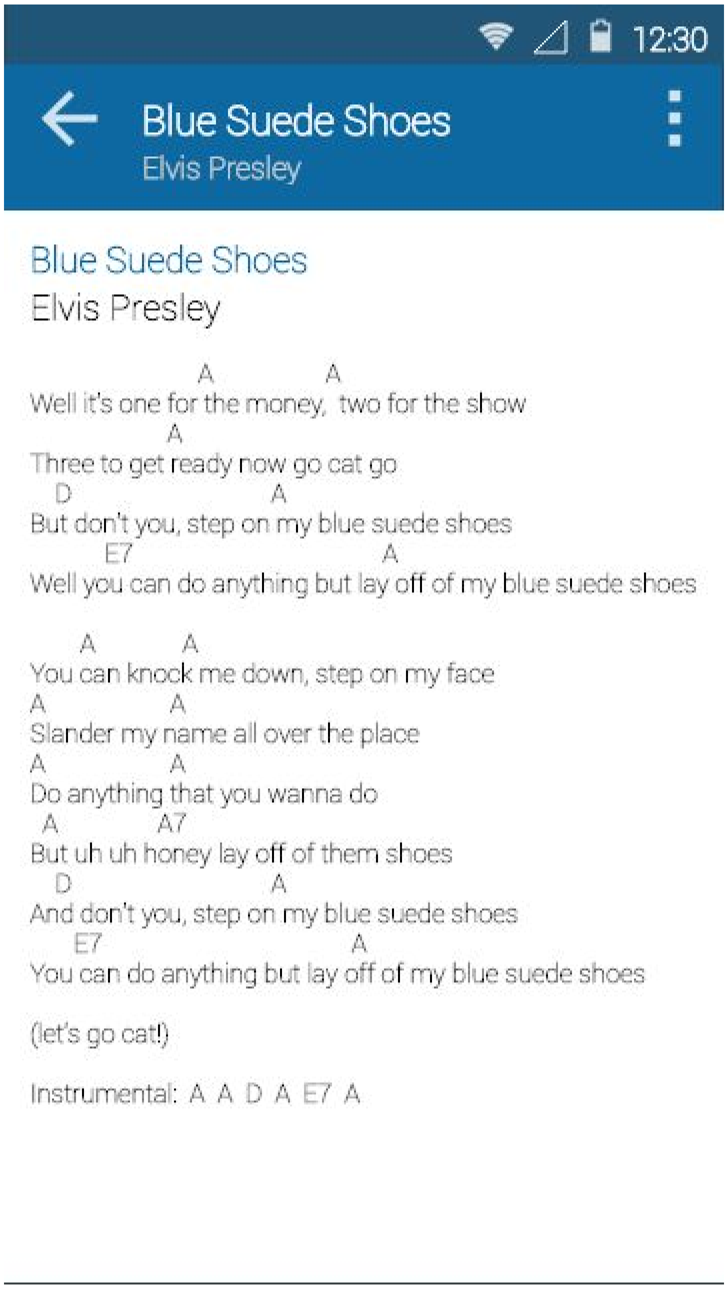
Add to library

Clocks
Coldplay
D - 120 ♩
Modified on Dec 13, 2015

Come Together
The Beatles
Bm - 120 ♩
Modified on Dec 23, 2015

Horse with no name
America
Em - 150 ♩
Modified on Dec 23, 2015

11. Song (chordpro format)



Blue Suede Shoes
Elvis Presley

```
                A           A
Well it's one for the money,  two for the show
             A
Three to get ready now go cat go
   D                  A
But don't you, step on my blue suede shoes
        E7                      A
Well you can do anything but lay off of my blue suede shoes

     A         A
You can knock me down, step on my face
A            A
Slander my name all over the place
A            A
Do anything that you wanna do
  A          A7
But uh uh honey lay off of them shoes
   D                  A
And don't you, step on my blue suede shoes
     E7                       A
You can do anything but lay off of my blue suede shoes

(let's go cat!)

Instrumental:  A  A  D  A  E7  A
```

Key Considerations

How will your app handle data persistence?

Most of the data will be stored in a SQLite database and delivered to the application through a content provider. The documents will be saved as files in the application internal storage.

Describe any edge or corner cases in the UX.

One important thing about this application is that it's have to be fast and reliable. Since its purpose is to be used in live situations for the only benefit of the musician (who wants to have his chart up and MIDI messages sent immediately), having fancy animations between views are not necessary and not planned to be implemented.

Describe any libraries you'll be using and share your reasoning for including them.

com.google.android.gms:play-services-auth - For Single Sign-In authentication
com.squareup.picasso:picasso - For image caching
com.github.kaaes:spotify-web-api-android - For spotify artist/song search
com.github.barteksc:android-pdf-viewer - For viewing PDFs

Out of the box android.midi library !

Describe how you will implement Google Play Services or other external services.

Google sign-in will be used, although it must be determined if the new Firebase platform is sufficient and maybe a better option (or a combination of both).

External service (repository and Rest API) will either be developed in Python (Django) or C# (WebAPI) and hosted on Azure.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Find, test and configure libraries for :
  - Search for an artist picture
  - Authenticate with a google account
  - Parse chordpro documents
  - Send MIDI messages
  - File pickering
  - Drag and drop
  - Display PDFs
- Determine a project structure (activities, fragments, layouts, menu, etc.)
- Upgrade to the latest Android Studio, gradle, AppCompat versions, etc.

Task 2: Create database objects

- Create database contract
- Create database
    - Create artist table
    - Create band table
    - Create document table
    - Create musician table
    - Create song table
    - Create setlist table
    - Create setlist_song table
    - Create message table
    - Create document_message table
- Create cursor loaders
- Create content provider
- Create parcelables

Task 3: Create Main activity objects

- Create drawer layout and menu
- Create main activity layout and AppCompatActivity

Task 4: Create Library fragment objects

- Create Library layout and Fragment (with TabLayout)
- Create Artists tab layout and Fragment
- Create Add/Edit Artist layout and Activity
- Implement Spotify search for the artist picture
- Create Songs tab layout and Fragment
- Create Add/Edit Song layout and Activity
- Create Document Add/Edit layout and Activity
- Create FilePicker layout and Activity for choosing document
- Create Document Viewer Activity (for Text, PDF, Image or Chordpro formats)
- Create Sets tab layout and Fragment
- Create Add/Edit Set layout and Activity
- Create Set details layout and Activity (with reorder)
- Create SongPicker layout and Activity for choosing songs (multi-select)
- Create Multiple Document Viewer Activity (with swiping)
- Create context menus and toolbar menus for all activities and fragments
- Implement Song editing (for text and chordpro formats)
- Implement Document Export

Task 5: Create Band fragment objects

- Create Band layout and Fragment
- Create Add/Edit band layout and Activity
- Create Members layout and Activity
- Create Add/Edit Member layout and Activity
- Implement send email to Band, Member
- Create context menus and toolbar menus for all activities and fragments

Task 6: Create MIDI fragment objects

- Create Midi layout and Fragment
- Create Add/Edit MIDI message layout and Activity
- Implement send MIDI message
- Create context menus and toolbar menus for all activities and fragments

Task 7: Create back-end database

- Create database
    - Create user table
    - Create artist table
    - Create band table
    - Create document table
    - Create musician table
    - Create song table
    - Create setlist table
    - Create setlist_song table
    - Create message table
    - Create document_message table
- Create UnitOfWork and database context (.NET)

Task 8: Create RESTFul services

- Implement CRUD operations on all tables
- Implement OAuth2 authentication
- Provide a basic out of the box UI for executing requests online

Task 9: Implement synchronization

- Implement synchronization through an IntentService
- Synchronize on demand and on application start
- Push changes as they occur

- Implement simple replace instead of merge, based on datetime stamping

Task 10: Create Community fragment objects

- Create Community layout and Fragment (with tabLayout)
- Create Connected tab layout and Fragment
- Create Requests tab layout and Fragment
- Create context menus and toolbar menus for all activities and fragments

Task 11: Create Search activity objects

- Create Search layout and Fragment (with tabLayout)
- Create Search for People tab layout and Fragment
- Create Search for Song tab layout and Fragment
- Create context menus and toolbar menus for all activities and fragments

Task 12: Create Settings activity objects

- Create Settings layout and Activity
- Settings
    - Theme (Material Dark / Material Light)
    - Backup
    - Import batch
    - Restore
    - Always on
    - Backlight
    - Etc.

Task 13: Implement login and authentication

- Implement Google Sign-In
- Implement Logout

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"

- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"